



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"
РТУ МИРЭА

Институт Информационных Технологий
Кафедра Вычислительной Техники

ПРАКТИЧЕСКАЯ РАБОТА №1

по дисциплине
«Теория принятия решений»
Метод Парето

Студент группы: ИКБО-42-23

Голев С. С.
(Ф. И. О. студента)

Преподаватель

Железняк Л.М.
(Ф.И.О. преподавателя)

Москва 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 МЕТОД ПАРЕТО.....	4
1.1 Выбор Парето-оптимального множества.....	4
1.2 Указание верхних/нижних границ критериев.....	5
1.3 Субоптимизация.....	6
1.4 Лексикографическая оптимизация.....	7
1.5 Результаты работы программы.....	7
ЗАКЛЮЧЕНИЕ.....	8
ПРИЛОЖЕНИЯ.....	10

ВВЕДЕНИЕ

Метод Парето, или принцип Парето, широко используется в задачах многокритериального принятия решений и оптимизации. Основная идея метода заключается в следующем:

Сравнение альтернатив по нескольким критериям:

При наличии множества вариантов (альтернатив) решения сравниваются по ряду критериев. Один вариант считается доминирующим по отношению к другому, если он по каждому критерию не хуже, а хотя бы по одному – лучше.

Парето-оптимальное множество:

Решения, которые не доминируются ни одним другим, называются Парето-оптимальными. Это означает, что нельзя улучшить хотя бы один критерий, не ухудшив при этом другой. Такое множество представляет собой набор лучших компромиссных вариантов, из которого затем может быть выбран окончательный вариант с учётом дополнительных предпочтений или ограничений.

1 МЕТОД ПАРЕТО

Задается множество альтернатив и их критериев со стремлениями.

Каждая альтернатива сравнивается с другими. Если найдется такая альтернатива s_j , что $s_j \succ s_i$ (то есть s_j не хуже s_i по всем критериям и хотя бы по одному критерию лучше), то s_i считается доминируемой, а s_j доминирующей.

Если для s_i ни одна другая альтернатива не доминирует над ней, s_i считается Парето-оптимальной.

Собираем все альтернативы, которые не доминируются ни одной другой, в множество, которое является Парето-оптимальным множеством.

1.1 Выбор Парето-оптимального множества

Предметная область: Выбор космического корабля.

Таблица 1 – Альтернативы со стремлениями

	Космический корабль	Цена (кредиты) (-)	Скорость (км/ч) (+)	Время входа в гиперпространство (сек) (-)	Количество орудий (шт) (+)	Мощность щитов (Вт) (+)
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	S-100	21000	4800	3.1	3	150
4	F-T2	30000	5100	4.0	3	110
5	CR90	25000	4600	3.5	2	130
6	IL-5	26000	4700	3.7	2	100
7	FT-6	35000	4400	4.5	2	100
8	FT-8	34000	4500	4.3	3	115
9	S-13	33000	4600	4.1	2	105
10	S-SC4	32000	4700	3.9	3	125

Найдём оптимальное множество:

Таблица 1.2 – Нахождение оптимального множества

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5			3							
6	1	2	3							
7		2	3	4	5					
8		2	3							
9		2	3	4	5					
10			3							

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

1.2 Указание верхних/нижних границ критериев.

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Для вариантов решений зададим границы. Цена корабля не должна превышать 22000 кредитов, скорость не менее 4800 км/ч, время входа в гиперпространство не менее 3.0 секунд, количество орудий 3 и более, мощность щитов больше 100 ватт.

Таблица 2 – Парето-оптимальные варианты с указанием границ

	Космический корабль	Цена (кредит ы) (-)	Скорость (км/ч) (+)	Время входа в гиперпрос транство (сек) (-)	Количество орудий (шт) (+)	Мощность щитов (Вт) (+)
1	TZ-24	22000	4900	3.2	4	120
2	S-100	21000	4800	3.1	3	150

1.3 Субоптимизация

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Главный критерий: Скорость.

Ограничения: Цена не более 25000 кредитов, время входа в гиперпространство больше 3.2 секунда, количество орудий больше 2, мощность щитов не менее 120 ватт.

Таблица 3 – Парето-оптимальные варианты

	Космический корабль	Цена (кредит ы) (-)	Скорость (км/ч) (+)	Время входа в гиперпрос транство (сек) (-)	Количество орудий (шт) (+)	Мощность щитов (Вт) (+)
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	CR90	25000	4600	3.5	2	130
4	S-100	21000	4800	3.1	3	150
5	F-T2	30000	5100	4.0	3	110

Самый лучший корабль: TZ-24

1.4 Лексикографическая оптимизация

Упорядочим критерии по приоритету: важнейший критерий - Скорость, следующие: количество орудий, мощность щитов, цена, время входа в гиперпространство.

Таблица 4 – Альтернативы со стремлениями

	Космический корабль	Цена (кредиты) (-)	Скорость (км/ч) (+)	Время входа в гиперпространство (сек) (-)	Количество орудий (шт) (+)	Мощность щитов (Вт) (+)
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	S-100	21000	4800	3.1	3	150
4	F-T2	30000	5100	4.0	3	110
5	CR90	25000	4600	3.5	2	130
6	IL-5	26000	4700	3.7	2	100
7	FT-6	35000	4400	4.5	2	100
8	FT-8	34000	4500	4.3	3	115
9	S-13	33000	4600	4.1	2	105
10	S-SC4	32000	4700	3.9	3	125

Самый лучший корабль: F-T2.

1.5 Результаты работы программы

```
Оптимальное множество:  
['TIE Fighter', 'TZ-24', 'S-100', 'F-T2', 'CR90']  
  
Указание верхних/нижних границ критериев:  
['TZ-24', 'S-100']  
  
Субоптимизация:  
['TZ-24']  
  
Лексикографическая оптимизация:  
F-T2
```

Рисунок 1 – Результат работы программы.

ЗАКЛЮЧЕНИЕ

В данной работе был изучен алгоритм Парето, выполнен ручной расчёт по этому алгоритму, написана программа на языке Python, которая реализует данный алгоритм, а также изучены улучшения этого алгоритма указания верхних и нижних границ критериев, субоптимизация, лексикографическая оптимизация.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

ПРИЛОЖЕНИЯ

Приложение А – Код реализации метода Парето на языке Python.

Приложение А

Код реализации метода Парето на языке Python.

Листинг А.1. . Реализация Парето.

```
import pandas as pd

alts = pd.DataFrame([
    {"name": "TIE Fighter", "credits": 20000, "speed": 5000, "hyper": 3.0,
    "weapons": 2, "shields": 100},
    {"name": "TZ-24", "credits": 22000, "speed": 4900, "hyper": 3.2, "weapons":
4, "shields": 120},
    {"name": "S-100", "credits": 21000, "speed": 4800, "hyper": 3.1, "weapons":
3, "shields": 150},
    {"name": "F-T2", "credits": 30000, "speed": 5100, "hyper": 4.0, "weapons":
3, "shields": 110},
    {"name": "CR90", "credits": 25000, "speed": 4600, "hyper": 3.5, "weapons":
2, "shields": 130},
    {"name": "IL-5", "credits": 26000, "speed": 4700, "hyper": 3.7, "weapons":
2, "shields": 100},
    {"name": "FT-6", "credits": 35000, "speed": 4400, "hyper": 4.5, "weapons":
2, "shields": 110},
    {"name": "FT-8", "credits": 34000, "speed": 4500, "hyper": 4.3, "weapons":
3, "shields": 115},
    {"name": "S-13", "credits": 33000, "speed": 4600, "hyper": 4.1, "weapons":
2, "shields": 105},
    {"name": "S-SC4", "credits": 32000, "speed": 4700, "hyper": 3.9, "weapons":
3, "shields": 125},
])

min_crit = ["credits", "hyper"]
plus_crit = ["speed", "weapons", "shields"]

def dom(a, b):
    crits = sum(a[min_crit].values <= b[min_crit].values) +
sum(a[plus_crit].values >= b[plus_crit].values)

    return crits == len(min_crit) + len(plus_crit)

def pareto(alts):
    pareto = []
    for i in range (len(alts)):
        for j in range (i+ 1,len(alts)):
            if dom(alts.iloc[i], alts.iloc[j]):
                pareto.append(alts.iloc[i])
                break

    return pd.DataFrame(pareto)

opt_alt = pareto(alts)

print("\nОптимальное множество:")
print(opt_alt["name"].to_list())
```

Продолжение листинга A.1.

```
fil_opt_alt = opt_alt[
    (opt_alt["credits"] <= 22000) &
    (opt_alt["speed"] >= 4800) &
    (opt_alt["hyper"] >= 3.0) &
    (opt_alt["weapons"] >= 3) &
    (opt_alt["shields"] > 100)
]

print("\nУказание верхних/нижних границ критериев:")
print(fil_opt_alt["name"].to_list())

fil_opt_alt = opt_alt[
    (opt_alt["credits"] <= 25000) &
    (opt_alt["hyper"] >= 3.2) &
    (opt_alt["weapons"] > 2) &
    (opt_alt["shields"] >= 120)
]

fil_opt_alt = fil_opt_alt.sort_values(by="speed", ascending=False)

print("\nСубоптимизация:")
print(fil_opt_alt["name"].to_list())

criteria_priority = ["speed", "weapons", "shields", "credits", "hyper"]

filtered_opt_alt = alts.sort_values(by=criteria_priority, ascending=[False, False,
False, True, True])

print("\nЛексикографическая оптимизация:")
print(filtered_opt_alt["name"].to_list()[0])
```

Конец листинга A.1.